Model-based fault-tolerant control with robustness to unanticipated faults [⋆]

Laurentz E. Olivier [*] **and Ian K. Craig** [**]

[*] *Department of Electrical, Electronic, and Computer Engineering,*
University of Pretoria, Pretoria, South Africa,
(e-mail: laurentz.olivier@sasol.com)
[**] *Department of Electrical, Electronic, and Computer Engineering,*
University of Pretoria, Pretoria, South Africa,
(e-mail: ian.craig@up.ac.za)

Abstract: Fault-tolerant control is important for the autonomous operation of complex processes. When model predictive control is used, fault detection and diagnosis is often based on the available process model used by the controller. Unanticipated faults can however cause misdiagnosis of faults, and consequently incorrect compensation actions. A fault-tolerant model predictive controller is presented in this article and tested on a grinding mill circuit simulator. The fault diagnosis algorithm quickly and accurately detects anticipated faults based on the generalized likelihood ratio test. Unanticipated faults are isolated when the process data do not sufficiently match the most probable anticipated fault data. The scheme is applicable to nonlinear multiple-input multiple-output systems.

Keywords: Fault diagnosis, generalized likelihood ratio, model-based control, nonlinear control, grinding mill circuit control.

1. INTRODUCTION

Automation is an important tool for modern processing plants to operate profitably under increasingly stringent environmental and safety regulations. If a failure occurs on the processing plant (including failure of a piece of processing equipment, an actuator, or a measurement device) the plant operating performance will likely decrease (Zhang and Jiang, 2008), and such faults easily lead to production stoppages (Blanke et al., 1997). Fault-tolerant control (FTC), which is specifically concerned with the ability to regulate the plant when faults occur, can generally alleviate the effects of faults, and may be able to prevent plant stoppages.

FTC is broadly classified as being either passive or active (Zhang and Jiang, 2008). Passive FTC has the objective to design the controller to be robust against a class of presumed faults. Active FTC aims to isolate faults (the source and sometimes the magnitude) and then to adapt the control strategy such that the stability and control performance of the system might be maintained. Active FTC relies heavily on real-time fault detection and diagnosis (FDD) algorithms to provide accurate, and up-to-date information about the true system status. FDD algorithms can be classified as either being data driven or model-based, depending on the process knowledge that is required *a priori* (Venkatasubramanian et al., 2003).

Model predictive control (MPC) is the most widely used advanced control strategy in the process industry (Bauer and Craig, 2008). The implementation of a successful MPC requires a sufficiently accurate process model. Using FTC along with MPC therefore makes model-based FDD a logical choice. Model-based fault diagnosis often relies on the analysis of residuals (Alcorta Garcia and Frank, 1997), which may generally be regarded as an indication of the difference between the expected plant response and the measured plant response. The expected plant response is commonly supplied by a state observer, and Deshpande et al. (2009) regards the design of the state observer as the key to successful integration of fault tolerance with predictive control.

The generalized likelihood ratio (GLR) method for fault diagnosis presented in Deshpande et al. (2009), is based on comparing the measured plant response with those from a group of observers, each designed with a different fault presumed. The fault presumed by the observer that best explains the plant measurements is chosen as the fault that has occurred. An observer has to be designed for each anticipated fault, and the method can therefore not directly handle novel (or unanticipated) faults. Novel identifiability is listed by Venkatasubramanian et al. (2003) as one of the ideal characteristics of a fault diagnosis system.

Farrell et al. (1993) notes that unanticipated faults may trigger the fault detection test, but does not match any of the expected faulty behaviours (of the anticipated faults). This unanticipated fault may then affect the system in a number of different ways. Firstly there may be very little impact on the process. This is the ideal case, and is often handled by setting up the fault detection test to not be triggered in this case. Secondly this fault may trigger the false isolation of an anticipated fault. The controller

[⋆] This work is based on the research supported in part by the National Research Foundation of South Africa (Grant Number 90533).

reaction to such a false diagnosis may be worse than with no diagnosis, and therefore this situation should be avoided. Lastly the fault could remain unaccommodated. Farrell et al. (1993) addressed how the effects of such an unanticipated fault may be offset for aircraft control by representing the unanticipated fault as an unmeasured force on the body of the aircraft. This approach is helpful, but is not generally expandable for process control.

Venkatasubramanian et al. (2003) notes that there is usually enough process data available to characterize nominal behaviour, but generally not enough to satisfactorily characterize abnormal operating regions (such as with faults). It is for this reason that some form of relation of how the fault affects the plant is useful. Wan et al. (2013) used signed digraphs to determine the root cause of novel faults, where novel faults are defined as faults for which historical data are not available. The digraphs are set up according to operational knowledge of how the novel faults affect the plant, which is often available.

A modelling approach is presented by Yen and Ho (2004), where an artificial neural network is used as an online estimator of the unknown failure dynamics of a single-input single-output system. There does however not seem to be much literature available regarding handling unanticipated faults for multiple-input multiple-output systems within a model-based control framework.

In this work, the nonlinear GLR method is used to diagnose faults. Fault detection is completed first. Anticipated faults, for which fault mode observers are set-up, are then identified when the corresponding fault mode observer sufficiently matches the measured plant outputs. If no fault mode observer sufficiently matches the measurements, an unanticipated fault is assumed, the effects of which are subsequently dealt with via feedback as performed by the MPC.

This method is illustrated through simulation of a nonlinear run-of-mine ore milling circuit, for which various faults are introduced, but fault-mode observers are only defined for some faults.

2. FAULT DETECTION AND ISOLATION

Most methods for model-based fault detection and diagnosis are based on residual analysis (see e.g. Alcorta Garcia and Frank (1997)). In this work the nonlinear version of the GLR method (as described by Deshpande et al. (2009)) is used. In this section an overview of the nonlinear GLR method is given along with the fault representation expressions.

2.1 Fault detection

Consider the innovation sequence calculated from the outputs of the nominal state observer (refer to Fig. 1):

$$\gamma(k) = y(k) - \hat{y}(k), \tag{1}$$

with $k \in [t, t+N]$, and $\hat{y} = g(\hat{x}_k, \theta_k)$ the outputs generated from the estimated states. Without any faults, and with a model that sufficiently encapsulates the system dynamics, the innovation sequence is zero-mean Gaussian white noise. If however any faults are present the state estimates become biased (Deshpande et al., 2009) and the innovation sequence is no longer zero-mean Gaussian noise.

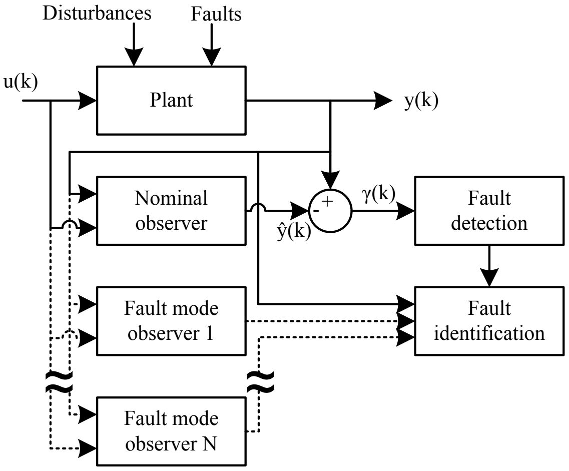

Fig. 1. Fault detection and isolation

Hotelling's T^2 statistic is an often-used metric for fault detection (Qin, 2012). The purpose of the statistic is to indicate a deviation in the data from normal operation, and can be expressed as:

$$T^2 = (x - \bar{x})^T V^{-1} (x - \bar{x}) \tag{2}$$

where x is the current observation with historical mean $\bar{x}$, and V is the covariance matrix. Akin to the T^2 statistic, a test statistic over an evaluation window, expressed as

$$\epsilon(t, N) = \sum_{k=t}^{t+N} \gamma(k)^T V(k)^{-1} \gamma(k) \tag{3}$$

is used to detect the fault. If this test statistic exceeds a threshold the fault is confirmed. The threshold and window length are tuning parameters for the fault detection test. A longer window length ensures more reliable but slower fault detection (because of its inherent filtering). The threshold is related to the severity of the fault as well as the window length.

2.2 Fault diagnosis

Suppose a set of observers is used, each operating with a different postulated fault, and each generates outputs along with the nominal observer (also indicated in Fig. 1). The problem of fault isolation is then finding the fault mode observer that best explains the measurement sequence $\{y(t) \ldots y(t+N)\}$ generated over the time window for which the fault was detected. The NL-GLR method can then be stated in mathematical form as:

$$\min_{b_{f_j}} \left(J_{f_j}\right) = \sum_{i=t}^{t+N} \gamma_{f_j}(i)^T V_{f_j}(i)^{-1} \gamma_{f_j}(i), \tag{4}$$

where $\gamma_{f_j}(i)$ and $V_{f_j}(i)$ are respectively the innovations and innovation covariance matrices generated by the fault mode observer corresponding to fault f_j. The isolated fault corresponds to the fault mode observer for which J_{f_j} is the smallest, with $\hat{b}_{f_j}$, the fault magnitude that produces this minimum value.

Because the innovation sequence of (1) and the innovation covariance matrix $(V(k))$ appear directly in the Kalman filtering framework, it is natural to make use of the Kalman filter as the state observer (or the extended Kalman filter (EKF) in the nonlinear case). The EKF however, as is often used in the nonlinear case, suffers some known limitations (see Julier and Uhlmann (2004) for a more complete discussion). It is for this reason that particle filtering is rather used to complete the estimation task in this work. A more complete discussion of particle filtering in the nonlinear GLR framework is given by Olivier and Craig (2016).

2.3 Representing faults

The fault diagnosis effort in this paper is limited to actuator errors for simplicity of illustration, although many other errors can also be represented in the GLR framework (see e.g. Olivier and Craig (2016)).

If the j-th actuator is stuck abruptly at time t then the corresponding plant input can be represented as:

$$u_{u_j}(k) = m(k) + \left[b_{u_j} - e_{u_j}^T m(k)\right] e_{u_j} \sigma(k-t) \qquad (5)$$

where m is the requested actuator value, b_{u_j} represents the constant value at which the j-th actuator is stuck, e_{u_j} is the fault vector with element j equal to one and all other elements equal to zero, and $\sigma(t)$ is the unit step function:

$$\sigma(t) = 0 \text{ if } t < 0;\ \sigma(t) = 1 \text{ if } t \geq 0. \qquad (6)$$

3. PREVENTING INCORRECT ANTICIPATED FAULT ISOLATION

When a significant, unanticipated fault enters the system (i.e. a fault for which no fault-mode observer is defined) the fault detection test will still be triggered because of the biasing effect on the residuals. During the fault diagnosis step, the fault-mode observer that best explains the output measurements is chosen as the candidate for the actual fault that occurred. If the candidate observer is the incorrect choice, such as in the case with an unanticipated fault, the residuals generated by this fault-mode observer will not be Gaussian noise, because the residuals will be biased by the effect of the unexplained fault.

It is now important to determine whether the candidate fault-mode observer sufficiently explains the measurements, or whether a significant bias is still present in the residuals. Again the test statistic of (3) is used, with the residuals based on the most optimal fault-mode observer as:

$$\epsilon^*(t,N) = \sum_{k=t}^{t+N} \gamma_{f^*}(k)^T V_{f^*}(k)^{-1} \gamma_{f^*}(k) \qquad (7)$$

where f^* represents the fault mode observer that minimizes J_{f_j} in (4). If the fault-mode observer sufficiently explains the measurements, this error statistic will have a low value. If the value of the test statistic is too large, it is likely that an unanticipated fault is affecting the system.

The threshold for not isolating an anticipated fault is a tuning value, and generally has to be more strict than the fault confirmation threshold. This is because even an incorrect fault-mode observer may reduce the residual error somewhat, in accordance with the optimization intent of (4). Unanticipated faults will therefore not be incorrectly diagnosed as one of the anticipated faults for which a fault-mode observer has been identified. Unanticipated faults are handled in line with passive FTC, where the controller rather tolerates faults. Maciejowski (1998) notes that constrained predictive control has an implicit degree of fault tolerance. This is because an unavailable actuator will likely saturate (become equal to the high or low limit) and the MPC will merely use the next available actuator to achieve the control objectives.

The implicit fault-tolerance and disturbance rejection capabilities inherent in an MPC controller are therefore deemed acceptable to deal with faults that are not anticipated to occur frequently enough, or for which the effects of non-diagnosis are not large enough to warrant the use of separate fault-mode observers.

4. FAULT-TOLERANT NONLINEAR MPC

This section briefly explains the fault-tolerant nonlinear MPC implementation. The discussion pertains to the general discrete time state-space representation of a dynamic system

$$x_k = f\left(x_{k-1}, u_{k-1}, \theta_{k-1}, v_{k-1}\right) \qquad (8)$$

$$y_k = g\left(x_k, \theta_k, e_k\right) \qquad (9)$$

where $x \in \mathbb{R}^n$ is the state vector and $y \in \mathbb{R}^m$ is the output vector, $f(\cdot)$ and $g(\cdot)$ are possibly nonlinear functions describing the state transitions and the outputs respectively, u_k contains the exogenous inputs, θ_k represents the parameters, v_k is the state noise and e_k is the measurement noise.

The objective of a model predictive controller at each sampling instant is to minimise the scalar performance index

$$\min_u \Phi(u, x_k) \qquad (10)$$

$$s.t.\ x \in X, u \in U \qquad (11)$$

$$\theta_c(x,u) \leq 0 \qquad (12)$$

where $x : \mathbb{R} \to \mathbb{R}^{n_x}$ is the state trajectory, $u : \mathbb{R} \to \mathbb{R}^{n_u}$ is the control trajectory, x_k is the state at time step k and $\theta_c(x,u)$ is the constraint vector.

The performance index (or objective function) to be minimized penalizes output values different from the reference values as well as excessive control moves. The objective function used in this work is similar to that shown in Qin and Bagwell (2003) as:

$$\Phi(\cdot) = \sum_{i=1}^{N_p} \left(\|y_{r,i} - y_i\|_{Q_r}^2 + Q_l y_i\right) + \sum_{i=0}^{N_c-1} \|\Delta u_i\|_R^2 \qquad (13)$$

where N_p and N_c are the prediction and control horizons respectively; $\|\cdot\|_Q$ represents the Q-weighted 2-norm; Q_r, Q_l, and R are weighting matrices corresponding to the reference tracking, linear optimization objectives, and control movements; y_r is the output reference and y is the output prediction.

The only difference in this formulation between the linear and nonlinear versions of the MPC is whether the output predictions are supplied by propagating the control vector through a linear or nonlinear model. The state estimates required by the MPC are also provided by means of a particle filter.

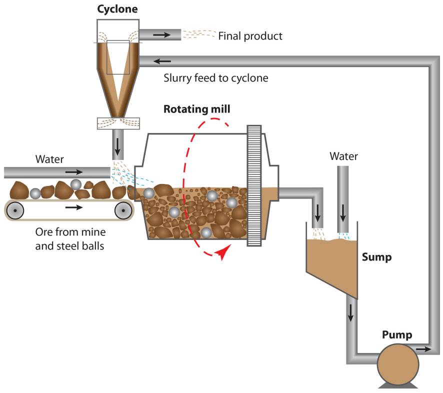

Fig. 2. Grinding mill circuit

5. APPLICATION EXAMPLE

The efficacy of the method is illustrated through application on a milling circuit simulator. A full nonlinear description of the plant is used by the NMPC, with faults being diagnosed through the nonlinear GLR method.

5.1 Process description

Only a brief description is given here of the grinding mill circuit. A much more complete description can be found in Le Roux et al. (2013), from which all of the parameter values listed in Table 1 can also be obtained.

The layout of the milling circuit is shown in Fig. 2. Ore from the mine is added to the grinding mill along with steel balls and water. Ore is ground down into fine particles inside the mill, and exits as a slurry through an end-discharge grate. Note that coarse ore and steel balls cannot pass through the grate. The slurry is sent to a sump where further water is added before being pumped to a cyclone for classification. Sufficiently ground down material leaves the top of the cyclone as the product of the milling circuit. Material that should be ground down further leaves the bottom of the cyclone and re-enters the mill.

The inputs into the milling circuit are the mill water feed (MIW), mill solids feed (MFS), mill steel balls feed (MFB), the sump water feed (SFW), and the cyclone feed flow-rate (CFF). These are all manipulated variables in the MPC. The milling circuit outputs are the mill load (LOAD), sump volume (SVOL), particle size estimate (PSE), circuit throughput (THP), the mill power draw (P_{mill}), and the cyclone feed density (CFD). The MPC controlled variables are the PSE, LOAD, and THP. The other measurements are used for fault detection and diagnosis.

The mill state equations are given by:

$$\dot{X}_{mw} = MIW - V_{mwo} \tag{14}$$

$$\dot{X}_{ms} = (1-\alpha_r)\frac{MFS}{D_s} - V_{mso} + RC \tag{15}$$

$$\dot{X}_{mf} = \alpha_f \frac{MFS}{D_s} - V_{mfo} + FP \tag{16}$$

$$\dot{X}_{mr} = \alpha_r \frac{MFS}{D_s} - RC \tag{17}$$

$$\dot{X}_{mb} = \frac{MFB}{D_b} - BC. \tag{18}$$

with

$$V_{mwo} = V_V \cdot \varphi \cdot X_{mw} \left(\frac{X_{mw}}{X_{mw} + X_{ms}} \right) \tag{19}$$

$$V_{mso} = V_V \cdot \varphi \cdot X_{mw} \left(\frac{X_{ms}}{X_{mw} + X_{ms}} \right) \tag{20}$$

$$V_{mfo} = V_V \cdot \varphi \cdot X_{mw} \left(\frac{X_{mf}}{X_{mr} + X_{ms}} \right) \tag{21}$$

$$BC = \frac{1}{D_b \phi_b} \cdot P_{mill} \cdot \varphi \cdot \left(\frac{X_{mr}}{X_{mr} + X_{ms}} \right) \tag{22}$$

$$RC = \frac{1}{D_s \phi_r} \cdot P_{mill} \cdot \varphi \cdot \left(\frac{X_{mr}}{X_{mr} + X_{ms}} \right) \tag{23}$$

$$FP = \frac{P_{mill}}{D_s \phi_f \left[1 + \alpha_{\phi_f} \left(\frac{LOAD}{v_{mill}} - v_{P_{max}} \right) \right]} \tag{24}$$

$$\varphi = \left(\frac{\max\left[0, \left(X_{mw} - \left(\frac{1}{\varepsilon_{ws}} - 1 \right) X_{ms} \right) \right]}{X_{mw}} \right)^{0.5} \tag{25}$$

$$P_{mill} = P_{max} \cdot \{1 - \delta_{P_v} Z_x^2 - \delta_{P_s} Z_r^2\} \cdot (\alpha_{speed})^{\alpha_P} \tag{26}$$

$$Z_x = \frac{LOAD}{v_{P_{max}} \cdot v_{mill}} - 1 \tag{27}$$

$$Z_r = \frac{\varphi}{\varphi_{P_{max}}} - 1. \tag{28}$$

The sump state equations are:

$$\dot{X}_{sw} = V_{mwo} + SFW - V_{swo} \tag{29}$$

$$\dot{X}_{ss} = V_{mso} - V_{sso} \tag{30}$$

$$\dot{X}_{sf} = V_{mfo} - V_{sfo}; \tag{31}$$

$$V_{swo} = CFF \cdot \left(\frac{X_{sw}}{X_{ss} + X_{sw}} \right) \tag{32}$$

$$V_{sso} = CFF \cdot \left(\frac{X_{ss}}{X_{ss} + X_{sw}} \right) \tag{33}$$

$$V_{sfo} = CFF \cdot \left(\frac{X_{sf}}{X_{ss} + X_{sw}} \right). \tag{34}$$

The cyclone is described as:

$$V_{ccu} = (V_{sso} - V_{sfo}) \cdot \left(1 - 0.6\, e^{-\frac{CFF}{\varepsilon_c}} \right) \cdot \left(1 - \left[\frac{F_i}{0.7} \right]^4 \right) \cdot (1 - P_i^4) \tag{35}$$

$$V_{cwu} = V_{swo} \cdot \frac{V_{ccu} - F_u \cdot V_{ccu}}{F_u \cdot V_{swo} + F_u \cdot V_{sfo} - V_{sfo}} \tag{36}$$

$$V_{cfu} = V_{sfo} \cdot \frac{V_{ccu} - F_u \cdot V_{ccu}}{F_u \cdot V_{swo} + F_u \cdot V_{sfo} - V_{sfo}} \tag{37}$$

$$F_u = 0.6 - (0.6 - F_i) \cdot e^{-\frac{V_{ccu}}{\alpha_{su} \varepsilon_c}}, \tag{38}$$

$$F_i = \frac{V_{sso}}{V_{swo} + V_{sso}} \tag{39}$$

$$P_i = \frac{V_{sfo}}{V_{sso}}. \tag{40}$$

with $V_{cfo} = V_{sfo} - V_{cfu}$ and $V_{cco} = (V_{sso} - V_{sfo}) - V_{ccu}$.

Table 1. Parameters and constants contained in the milling circuit equations.

Parameter	Value	Description
α_f	0.055	Fraction of fines in the ore
α_r	0.465	Fraction of rocks in the ore
ϕ_f	29.57	Power per ton of fines produced $[\text{kW}\cdot\text{h}/\text{t}]$
ϕ_r	6.03	Rock abrasion factor $[\text{kW}\cdot\text{h}/\text{t}]$
ϕ_b	90	Steel abrasion factor $[\text{kW}\cdot\text{h}/\text{t}]$
D_s	3.2	Feed ore density $[\text{t}/\text{m}^3]$
D_b	7.8	Steel ball density $[\text{t}/\text{m}^3]$
ε_{ws}	0.6	Maximum water-to-solids volumetric flow at zero slurry flow
V_V	84	Volumetric flow per "flowing volume" driving force $[\text{h}^{-1}]$
P_{max}	1661	Maximum mill motor power [kW]
δ_{P_v}	0.5	Power change parameter for volume of mill filled
δ_{P_s}	0.5	Power change parameter for fraction solids in the mill
$v_{P_{max}}$	0.34	Fraction of mill volume filled for maximum power
$\varphi_{P_{max}}$	0.57	Rheology factor for maximum mill power
α_P	1.0	Fractional power reduction per fractional reduction from maximum mill speed
v_{mill}	59.1	Mill volume $[\text{m}^3]$
α_{ϕ_f}	0.01	Fractional change in kW/fines produced per change in fractional filling of mill
ε_c	128.85	Coarse split parameter
α_{su}	0.87	Parameter related to solids in cyclone underflow

The milling circuit outputs are:

$$\begin{aligned} LOAD &= X_{mw} + X_{ms} + X_{mr} + X_{mb} \\ SVOL &= X_{sw} + X_{ss} \\ PSE &= \frac{V_{fo}}{V_{co}+V_{fo}} \\ THP &= V_{co} + V_{fo} \\ CFD &= \frac{X_{sw}+D_s X_{ss}}{X_{sw}+X_{ss}} \end{aligned} \tag{41}$$

as well as P_{mill} given in (26). The parameter values contained in the process equations are listed in Table 1.

The specific values used by the NMPC and particle filters for the simulation are as listed in Olivier and Craig (2016).

5.2 Simulation results

The simulation is run for a total of 5 hours, and is propagated at a sampling period of 10 seconds. Fault-mode observers are only defined for MIW and CFF actuator errors. This limited number is merely to keep the example simple.

Firstly one of the anticipated faults are introduced. After 0.8 hours the valve supplying MIW fails such that the feed water reduces to 0 m^3/h. One hour into the simulation the PSE setpoint is increased by 15 % to illustrate the reference tracking performance in the presence of the fault.

The fault is detected shortly after its introduction. The FDI selects an MIW fault as the most probable from the set of fault-mode observers. The residuals are calculated and the test statistic for isolating an anticipated fault (7) is smaller than the threshold. The MIW fault is therefore successfully diagnosed and the MPC continues to control with the fault information. Fig. 3 shows the measurements and output predictions from the MIW and CFF fault-mode observers over the fault diagnosis window. It can be confirmed visually that the predictions from the MIW fault-mode observer corresponds better with the predictions.

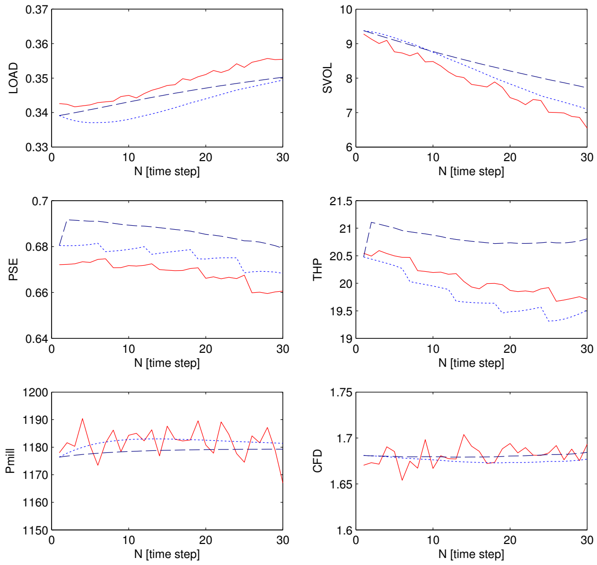

Fig. 3. Process measurements (red) and output predictions from MIW (light blue dotted) and CFF (dark blue dashed) fault-mode observers with an MIW fault present.

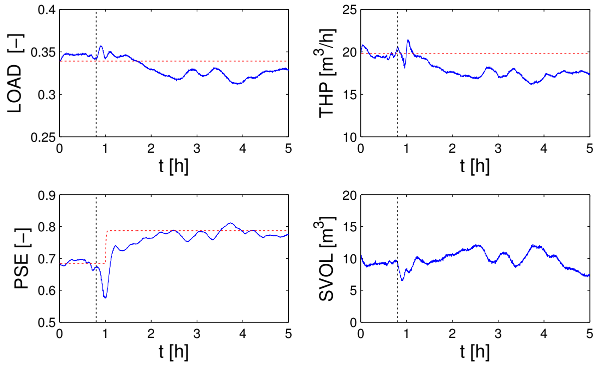

Fig. 4. Control performance with MIW fault.

Fig. 4 shows the control performance over the entire simulation run. The control performance is affected by the MIW error (especially PSE tracking), but recovery is swift owing to the quick and correct fault diagnosis. The introduction of the fault is indicated by the horizontal dashed line.

In the second simulation an unanticipated fault is introduced. After 0.8 hours the value of ϕ_f is decreased by 20 % to simulate a decrease in feed ore hardness. This large unmeasured disturbance biases the residuals to the extent that the fault detection test is triggered. The fault diagnosis algorithm executes and selects a CFF error as the candidate fault. The predictions (see Fig. 5) do however not sufficiently encapsulate the changes in the outputs to pass the threshold for confirming an anticipated fault. It is therefore assumed that an unanticipated fault has occurred, and the MPC is allowed to continue.

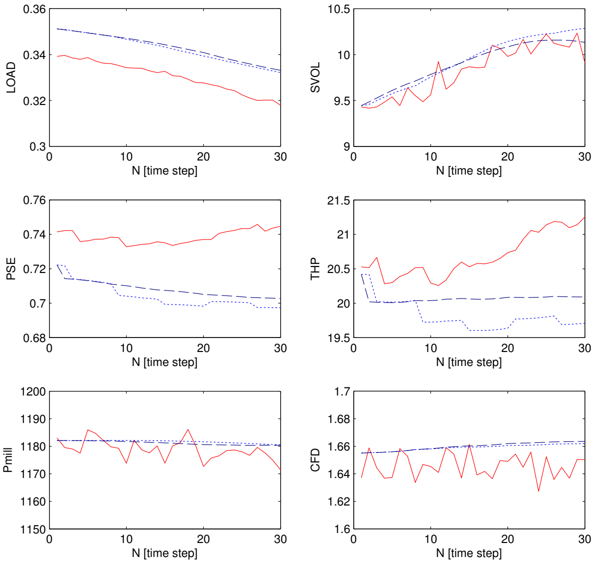

Fig. 5. Process measurements (red) and output predictions from MIW (light blue dotted) and CFF (dark blue dashed) fault-mode observers with a feed ore hardness disturbance.

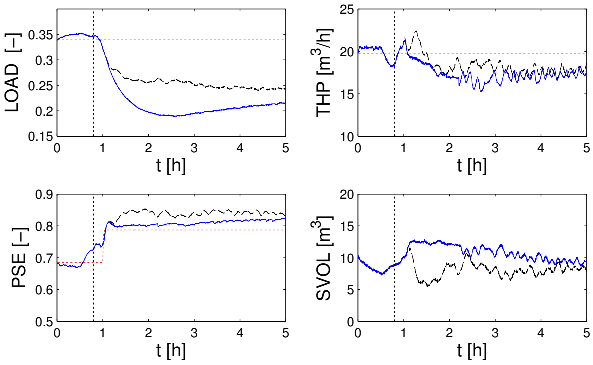

Fig. 6. Control performance with feed ore hardness disturbance without selecting an anticipated fault (blue) and when incorrectly selecting the CFF fault (black dashed).

Fig. 6 shows the control performance with the feed ore hardness disturbance (the introduction of which is indicated by the horizontal dashed line). The control performance (shown in blue) is better than if the CFF error was incorrectly diagnosed (shown in black); for similar throughputs the PSE tracking in blue is better.

6. CONCLUSION

Fault-tolerant MPC using the GLR method for FDD is presented in the paper. A bank of estimators provide the estimates for different presumed faults, from which the most likely fault is selected. The core contribution of this paper is the expansion of the method to not falsely diagnose an anticipated fault when an unanticipated fault occurs.

The method presented efficiently handles anticipated faults, and has the ability to not incorrectly diagnose faults in the presence of unanticipated faults. Successful application is illustrated through simulation of a run-of-mine ore milling circuit.

REFERENCES

Alcorta Garcia, E. and Frank, P.M. (1997). Deterministic nonlinear observer-based approaches to fault diagnosis: a survey. *Control Eng. Practice*, 5, 663 – 670.

Bauer, M. and Craig, I.K. (2008). Economic assessment of advanced process control - A survey and framework. *Journal of Process Control*, 18, 2 – 18.

Blanke, M., Izadi-Zamanabadi, R., Bøgh, S.A., and Lunau, C.P. (1997). Fault-tolerant control systems - a holistic view. *Control Eng. Practice*, 5, 693 – 702.

Deshpande, A.P., Patwardhan, S.C., and Narasimhan, S.S. (2009). Intelligent state estimation for fault tolerant nonlinear predictive control. *Journal of Process Control*, 19, 187 – 204.

Farrell, J., Berger, T., and Appleby, B. (1993). Using learning techniques to accommodate unanticipated faults. *IEEE Control Systems Magazine*, 13, 40 – 49.

Julier, S.J. and Uhlmann, J.K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92, 401 – 422.

Le Roux, J., Craig, I., Hulbert, D., and Hinde, A. (2013). Analysis and validation of a run-of-mine ore grinding mill circuit model for process control. *Minerals Engineering*, 43 - 44, 121 – 134.

Maciejowski, J.M. (1998). The implicit daisy-chaining property of constrained predictive control. *Applied Mathematics and Computer Science*, 8, 695 – 712.

Olivier, L.E. and Craig, I.K. (2016). Fault-tolerant nonlinear MPC using particle filtering. In *IFAC-PapersOnLine*, volume 49, issue 7, 177 – 182. URL http://ac.els-cdn.com/S2405896316304499/1-s2.0-S2405896316304499-main.pdf?_tid=175383cc-7a8d-11e6-903e-00000aab0f02&acdnat=1473865888_8537453cd0664cb128a0b1ccc95ed627.

Qin, S.J. (2012). Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36, 220 – 234.

Qin, S.J. and Bagwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Eng. Practice*, 11, 733 – 764.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S.N. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 27, 293 – 311.

Wan, Y., Yang, F., Lv, N., Xu, H., Ye, H., Li, W., Xu, P., Song, L., and Usadi, A.K. (2013). Statistical root cause analysis of novel faults based on digraph models. *Chemical Engineering Research and Design*, 91, 87 – 99.

Yen, G.G. and Ho, L.W. (2004). Intelligent on-line fault tolerant control of unanticipated catastrophic failures. *ISA transactions*, 43, 549 – 569.

Zhang, Y. and Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32, 229 – 252.